

(19)



(11)

EP 2 115 635 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention
of the grant of the patent:
12.06.2019 Bulletin 2019/24

(51) Int Cl.:
G06F 17/30 ^(2006.01) **G06F 17/22** ^(2006.01)
G06F 17/27 ^(2006.01)

(21) Application number: **07865690.7**

(86) International application number:
PCT/US2007/087582

(22) Date of filing: **14.12.2007**

(87) International publication number:
WO 2008/076887 (26.06.2008 Gazette 2008/26)

(54) AUTOMATIC SEARCH QUERY CORRECTION

AUTOMATISCHE SUCHANFRAGEKORREKTUR

CORRECTION AUTOMATIQUE DE REQUÊTES DE RECHERCHE

(84) Designated Contracting States:
**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR
HU IE IS IT LI LT LU LV MC MT NL PL PT RO SE
SI SK TR**

(30) Priority: **15.12.2006 US 875425 P**
12.01.2007 US 884839 P
16.04.2007 CN 200710101817

(43) Date of publication of application:
11.11.2009 Bulletin 2009/46

(73) Proprietor: **Google LLC**
Mountain View, CA 94043 (US)

(72) Inventor: **TAVOR, Guy**
Tel Aviv (IL)

(74) Representative: **Robinson, David Edward**
Ashdown et al
Marks & Clerk LLP
1 New York Street
Manchester M1 4HD (GB)

(56) References cited:
EP-A2- 0 967 555 KR-A- 970 007 686
KR-A- 20010 064 751 KR-A- 20060 118 253
US-A1- 2003 023 420 US-A1- 2004 260 536
US-A1- 2005 071 332

EP 2 115 635 B1

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Description

RELATED APPLICATIONS

[0001] This application claims priority to Chinese Patent Application No. 200710101817.8, filed on April 16, 2007, U.S. Provisional Patent Application Serial No. 60/875,425, filed on December 15, 2006, and U.S. Provisional Patent Application Serial No. 60/884,839, filed on January 12, 2007.

BACKGROUND

[0002] This specification relates to automatically changing a search query in a given script to a search query in another script.

[0003] A search engine enables a user to provide a search query for which search results are returned in response. In general, the query can include any character that the user is capable of generating with his or her input device. Input devices can have keys or buttons that are each capable of producing one of many characters. A keyboard, for example, can have multiple keyboard layouts. Each keyboard layout typically corresponds to the characters most frequently used within a particular writing system (i.e., a script). In countries whose writing systems are not based on the Latin alphabet, keyboards usually have at least two keyboard layouts, one for producing Latin characters (e.g., a U.S. English keyboard layout), the other for producing the characters of the user's native writing system (e.g., Cyrillic, Devanāgarī, or Arabic).

[0004] Generally, users are able to switch between two or more keyboard layouts very easily (e.g., typically with a single keystroke). The currently selected keyboard layout normally determines the characters produced by each key of the keyboard. A user may inadvertently use their input device to submit a query in one script when they really intended to provide the same query in another script. For example, a user searching for 'news' may inadvertently submit 'ד'קמ' on a Hebrew keyboard layout because the keystrokes that produce the characters 'ד'קמ' produce the characters 'news' on the corresponding Latin keyboard layout.

[0005] Reference is made to the following prior art:

EP 0 967 555 A2 <(MICROSOFT CORP [US]) 29 December 1999 (1999-12-29)> discloses converting text according to keyboard mode using language detection in a word processing environment.

US 2005/071332 A1 <(ORTEGA RUBEN ERNESTO [US] ET AL) 31 March 2005 (2005-03-31)> discloses correction of misspelled search terms.

SUMMARY

[0006] This specification discloses various examples of technologies for automatically changing a search query in a given script to a search query in another script. These examples feature methods, systems, apparatus, including computer program product apparatus. Each of these will be described in this summary by reference to the methods, for which there are corresponding systems and apparatus.

[0007] In general, in one aspect, a first search query is received from a user, the first search query having one or more characters, and a search result is determined based on the first search query. Based on the search result, the first search query is determined to have an incorrect input mapping. A first keyboard layout is identified for the first search query, and a second keyboard layout is identified. A corrected search query is generated from the first search query by mapping characters from the first keyboard layout to characters in the second keyboard layout. A corrected search result is determined based on the corrected search query, and the corrected search result is presented to the user.

[0008] Implementations may include one or more of the following features. Multiple corrected search queries can be generated; one of the multiple corrected search queries can be selected; and the corrected search result can be presented to the user based on the selected one of the multiple corrected search queries.

[0009] In general, in another aspect, a first search query is received from a user, the first search query having one or more characters, and a search result is determined based on the first search query. The first search query is determined to have an incorrect input mapping based on the search result, and a corrected search query is generated from the first search query from a dictionary of input mapping corrections. A corrected search result is determined based on the corrected search query, and the corrected search result is presented to the user.

[0010] In general, in another aspect, a first search query is received from a user, the first search query including at least one or a combination of two or more of (1) one or more characters, (2) user languages settings, (3) a user language preference, (4) a user interface language, (5) a geographic location, and (6) user search history. A search result is determined based on the first search query. The first search query is determined to have an incorrect input mapping based on the search result. A first keyboard layout is identified for the first search query. A second keyboard layout is identified. A corrected search query is generated from the first search query by mapping characters from the first keyboard layout to characters in the second keyboard layout. A corrected search result is determined based on the corrected search query. The corrected search result is presented to the user.

[0011] Implementations may include one or more of the following features. The user language preference can be associated with a part of an operating system's lan-

guage settings. The user language preference can be associated with information about a supplemental language support provided by an operating system. The user language preference can be associated with information of a web browser. The information of the web browser can be related to a language associated with the web browser. A page request of a web browser can be used to provide the user language preference.

[0012] The identifying of the first keyboard layout can be based on a language setting or language information of a system used by the user. The identifying of the first keyboard layout can be based on primary and secondary languages identified by the user. The identifying of the first keyboard layout can be based on a user preferred keyboard layout. The identifying of the first keyboard layout can be based on an inference from a language of the user interface. The identifying of the first keyboard layout can be based on information about a device connection of the user from which the user's geographic location and thus the user's most likely language are inferred. The identifying of the first keyboard layout can be based on search query history of the user. A portion of search results in the search query history, being less than an entire collection of search results in the search query history, can be used to identify the first keyboard layout. The portion of search results in the search query history can include search results that have been accessed by the user.

[0013] In general, in another aspect, a search query is received from a user, the search query including at least one or a combination of two or more of (1) one or more characters, (2) user languages settings, (3) a user language preference, (4) a user interface language, (5) a geographic location, and (6) user search history. A search result is determined based on the search query, and a keyboard layout associated with the search query is identified.

[0014] Implementations may include one or more of the following features. The identifying of the keyboard layout can be based on a language setting or language information of a system used by the user. The identifying of the keyboard layout can be based on primary and secondary languages identified by the user. The identifying of the keyboard layout can be based on a user preferred keyboard layout. The identifying of the keyboard layout can be based on an inference from a language of the user interface. The identifying of the keyboard layout can be based on information about a device connection of the user from which the user's geographic location and thus the user's most likely language are inferred. The identifying of the keyboard layout can be based on search query history of the user. A portion of search results in the search query history, being less than an entire collection of search results in the search query history, can be used to identify the first keyboard layout. The portion of search results in the search query history can include search results that have been accessed by the user.

[0015] In general, in another aspect, a search query is

received from a user, the search query including user search history; a search result is determined based on the search query; and a keyboard layout associated with the search query is identified.

[0016] Implementations may include one or more of the following features. The search query can include at least one of (1) one or more characters, (2) user languages settings, (3) a user language preference, (4) a user interface language, and (5) a geographic location. A portion of search results in the search query history, being less than an entire collection of search results in the search query history, can be used to identify the first keyboard layout. The portion of search results in the search query history can include search results that have been accessed by the user.

[0017] In general, in another aspect, a keyboard layout used by a user is determined to be incorrect based on first search results responsive to an initial search query submitted by the user.

[0018] Implementations may include one or more of the following features. Second search results can be provided to the user using a corrected search query. The corrected search query can be generated from a dictionary of input mapping corrections. A corrected search query can be generated from the initial search query by mapping characters from the keyboard layout used by the user to characters in a corrected keyboard layout. The corrected keyboard layout can be determined based on a search query history associated with the user. The corrected keyboard layout can be determined based on a language setting associated with the user. The corrected keyboard layout can be determined based on a geographic location associated with the user. Multiple corrected search queries can be provided to the user. Corrected search results can be provided to the user responsive to a selection of one of the multiple corrected search queries by the user.

[0019] These and other aspects and features, and combinations of them, may be expressed as methods, apparatus, systems, means for performing functions, program products, and in other ways.

[0020] Advantages of the aspects and features can include one or more of the following. Queries can be corrected automatically without the user having to correct the query manually. Users can provide queries using an unfamiliar keyboard layout and still receive results as though the query had been generated by the user's preferred keyboard layout.

[0021] The details of one or more examples are set forth in the accompanying drawings and the description below. Other features, aspects, and advantages of the invention will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0022]

FIG. 1 is a flow diagram of a process for detecting and correcting an incorrect query.

FIG. 2 illustrates two keyboard layouts used to produce two queries from the same keystrokes.

FIG. 3 illustrates query suggestions and each suggestion's associated query results.

Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0023] As shown in FIG. 1, a process 100 determines whether a search query has been mistakenly entered using an incorrect keyboard layout and, if so, corrects the search query. The search query is received from a user (step 110). The search query is a string of one or more characters. The characters of the search query correspond to the characters used in a particular script (e.g., Latin, Cyrillic, Hebrew or Arabic). Sometimes characters received from the user are not the characters the user intended to provide.

[0024] The process 100 includes receiving a search query in a particular script (step 110). Initial search results are determined based on the received query (step 120). Search results are determined based on any convenient mechanism for determining search results. Typically, a corpus of items will be searched to identify particular items that are relevant to the search query. In general, a search result identifies many items. However, the initial search result may include no search result items.

[0025] The process 100 determines whether the initial search query was likely generated using the correct keyboard layout (step 125). The items in the initial search result can be counted to determine whether the initial search result identified a sufficient number of items. In some implementations, a threshold can be used to determine whether the number of items is sufficient (e.g., greater than zero or greater than ten). In other implementations, the threshold is proportional to the total number of items in the corpus of items. For example, the threshold may be 25, if the corpus contains 10,000 items; however, the threshold may only be 2, if the corpus contains 1000 items.

[0026] If the initial search query was likely generated by the correct keyboard layout, then process 100 is complete when the initial search results are presented to the user (step 127). Alternatively, if the search query was likely generated by the incorrect keyboard layout, then the search query is automatically corrected as though it had been generated by a user using a target keyboard layout.

[0027] A target keyboard layout corresponds to the keyboard layout that the user most likely intended to use to provide the search query. A source keyboard layout corresponds to the keyboard layout that the user most

likely inadvertently used to provide the search query. The source and the target keyboard can be determined based on the characters used in the initial search query. Alternatively, the target keyboard can be determined based on a user's languages settings, language preferences, user interface language, geographic location, search history, or a combination of them.

[0028] The process 100 identifies the particular character set in which the search query is specified (step 130). In general, a character set refers to a group of characters within a family of characters. The family of characters is typically specified by a single consistent encoding (e.g., Unicode). The character set can be identified based on which group of characters are specified in the search query.

[0029] If the search query is specified in non-Latin characters (step 135), then the non-Latin search query can be corrected as though the user intended to generate the query on a Latin keyboard layout. In other words, the target keyboard layout is a Latin keyboard layout. The process 100 determines the source keyboard layout used to produce the non-Latin search query (step 140).

[0030] Many non-Latin character sets are produced by a single, predictable, keyboard layout. For example, Hebrew characters are generally produced using a standard Hebrew keyboard layout while Arabic characters are generally produced using a standard Arabic keyboard layout.

[0031] Not all non-Latin character sets, however, have a single corresponding keyboard layout. For example, Cyrillic script is used in both Russian and Bulgarian languages, and likewise, Cyrillic characters can be generated by both Russian and Bulgarian keyboard layouts. However, the Russian and Bulgarian keyboard layouts differ. The same keystroke on a Russian or Bulgarian keyboard layout will produce respectively different Cyrillic characters.

[0032] In general, a user provides a search query to a user interface typically associated with a particular language. When the user's source keyboard layout is ambiguous, the language of the search interface can be used to resolve the ambiguity. If, for example, the user has provided the search query to a Bulgarian user interface, the search query is likely to have been generated using a Bulgarian keyboard layout.

[0033] Alternatively, instead of resolving the ambiguity, query corrections can be generated for each applicable keyboard layout (e.g., for all applicable keyboard layouts or for a preferred subset). In some implementations, each corrected query is used to determine respective corrected search results. The numbers of items in the corrected search result are compared with one another to determine the source keyboard layout. Since a search query is unlikely to map to a meaningful term using an incorrect source keyboard layout, the corrected search result with the most items is taken as corresponding with the correct source keyboard layout.

[0034] If the search query is specified using Latin characters (step 135), then the Latin search query can be

corrected as though the user intended to generate the query on a non-Latin keyboard layout. In other words, the source keyboard layout is a Latin keyboard layout. Although there are multiple Latin keyboard layouts that vary somewhat, the most commonly used keys on any Latin keyboard generate largely the same characters (e.g., the character A-Z and a-z). The process 100 determines the most likely target keyboard layout (step 150).

[0035] In some implementations, all potential corrections can be generated for every possible target keyboard layout. The large number of potential keyboard layouts, however, means that such an approach is inefficient. Instead, the process 100 can determine the user's most likely target keyboard layout using one or a combination of five alternative approaches.

[0036] The first approach is to identify further language settings or information from the user's system. For example, the user's system can automatically provide (or be specifically queried to provide) information about the user's language preferences (e.g., operating system language settings). A user's language preference can be specified as (or derived from) a part of an operating system's language settings or information about supplemental language support provided by the operating system or web browser (e.g., installed languages or language packs). For example, a web browser can automatically provide a user's language preferences with a page request.

[0037] Using the second approach, the user can explicitly identify primary and secondary languages, or alternatively, user preferred keyboard layouts. For example, the user providing the search query is associated with a predetermined profile in which the user sets language preferences.

[0038] In the third approach, the language of the user interface is used to infer the user's language preferences. If the user provides the search query to a Russian user interface, then the target keyboard layout can be assumed to be a Russian keyboard layout.

[0039] Using the fourth approach, when the user provides a search request through a network connected device, information about the device connection is used to infer the user's geographic location and thus the user's most likely language. For example, the user's Internet protocol (IP) address can be used to identify the approximate geographic location (e.g., typically a particular country) of the user's network device. A particular country can be associated with one or more languages that a given user would likely be using.

[0040] The fifth approach considers a user's search query history. A user's search query history includes search queries provided previously by the user. Each search query in the search query history corresponds to one or more search results, some of which may have been accessed by the user (e.g., the user clicked on a search result). Each search result is associated with a language that ideally identifies the language of the search

result. The languages of the search results of all the user's search queries can be used to determine the most likely intended target language, from which the target keyboard layout is derived. For example, if, among all of a user's previous search results, 75% of search results are English and 20% are Hebrew, then the target keyboard layout is assumed to be English when the source keyboard layout is Latin. In some implementations, rather than consider all of the search results associated with the user's search query history, only the search results that have been accessed by the user are considered.

[0041] Each of the five approaches described above generally identifies a user's language preference. In general, a language that uses a non-Latin script has only one commonly-used keyboard layout. The particular keyboard layout that corresponds to the user's language preference is assumed to be the target keyboard layout.

[0042] The process 100 corrects the search query based on the identified source and target keyboard layouts (step 160). Keyboard layouts are well known and predefined. Therefore, a character mapping between keyboard layouts can be pre-defined. For example, a pre-defined mapping identifies that the Hebrew character 'נ' is produced on a Hebrew keyboard layout using the same keystroke that produces the Latin character 'n' on a Latin keyboard layout. A corrected search query can be generated by mapping each character in the received search query to a corresponding character in a corrected search query. A character in the received search query, generated by the source keyboard layout, is mapped to the corresponding character that would have been generated with the same keystroke using the target keyboard layout.

[0043] FIG. 2 illustrates two keyboard layouts, a US-English keyboard layout 200 and a Hebrew keyboard layout 250, used to produce two different queries based on the same keystrokes. The keyboard with the Latin keyboard layout 200 is used to produce a string 'news' 205. The sequence of characters 'n', 'e', 'w', and 's' are generated with the keystrokes shown in grey. For example, 'n' is generated from the keystroke 210.

[0044] The same keystrokes of the same keyboard with the Hebrew keyboard layout 250 generates the string 'נעש' 255. Note that although the string 255 appears to be reversed compared to the characters produced from the same sequence of keystrokes, Hebrew text is written right-to-left. Therefore, 'נ' is the first character generated by the keystroke 260, which is the same as keystroke 210 that generated 'n' using the Latin keyboard layout 200.

[0045] Returning to FIG. 1, the process 100 determines corrected search results based on the corrected search query (step 170). In some implementations, if more than one corrected search query is generated, then corrected search results are determined for each corrected search query.

[0046] The process 100 determines which of either the

initial search results or the corrected search results are presented to the user (step 180). In some implementations, the number of items in the initial search result is compared to the number of items in the corrected search result. The search result with the greatest number of items is presented to the user.

[0047] In some implementations, the initial and corrected search result are each a suggestion-set. The items of a suggestion-set identify query suggestions that a user can select as a new search query from which new search results may be generated. Typically, the suggestion-set is identified from a first corpus (e.g., a corpus of search query suggestions), while the new search results are identified from a second corpus (e.g., a corpus of web pages).

[0048] FIG. 3 illustrates two suggestion-sets that correspond to the two search queries, 'news' 300 and 'ד'ק'מ' 350 respectively. Each suggestion-set includes i suggestions (q_i) 310, wherein i may be any integer from 0 to n . Each suggestion is in itself a query that when submitted as a new search query to the second corpus, yields (or would be expected to yield) some number of results 320. For example, the first suggestion, q_0 is expected to generate x results. The value $n(q_i)$ 330 represents the expected number of results that the suggested query q_i would yield.

[0049] The information associated with each query suggestion can be used to determine which of the suggestion-sets (initial or corrected) is presented to the user. For example, the expected results of each query suggestion in a suggestion-set are used to score the suggestion-set. In some implementations, the suggestion-set is scored based on an average (or a maximum or standard deviation) of the expected results of each query suggestion. In other implementations, the suggestion-set is scored based on the median of the expected results of each query suggestion.

[0050] Alternatively, the score of a suggestion-set (qs_j) can be based on the following formula:

$$score_j = \left\lceil \frac{n(qs_j)}{S} \right\rceil$$

[0051] Where $n(qs_j)$ is the sum of $n(q_1)$, $n(q_2)$... $n(q_n)$ where $n(q_i)$ is the number of expected results for each potential search query q_i in the suggestion-set qs_j . The value of S can be the number of suggestions that is normally expected. In some implementations, the value of S is predefined (e.g., corresponding to a fixed limit for all suggestion sets). In other implementations, the value S can be periodically determined based on the average number of items in all suggestion-sets. The above formula demotes the scores of suggestion-sets that have fewer than the expected number of suggestions (S). In

some cases, the first corpus of query suggestions can include suggestions based on erroneous strings that have been entered using the wrong keyboard layout (e.g., a string that makes no sense except if mapped using another keyboard layout). The demotion of suggestion-set scores based on S prevents presenting a suggestion-set that refers only to the erroneous strings in the first corpus.

[0052] In general, the suggestion-set with the greatest score is presented to the user. In some implementations, the initial suggestion-set can be scored before correcting the initial search query. The score of the initial suggestion-set is used to determine whether the initial search results were likely generated using an incorrect keyboard layout. In other words, this scoring technique can be used rather than counting the items in the initial search results, which was described in reference to FIG. 1 and step 125).

[0053] In some implementations, when a search query is automatically corrected, information is presented to the user indicating the automatic correction. In some implementations, multiple potential search queries (e.g., when target or source keyboard layout is ambiguous) can be presented. One of these presented corrected search queries can be explicitly selected by a user.

[0054] In some implementations, a dictionary of common search query corrections can be maintained. The dictionary contains search query entries and corresponding corrections. The dictionary can be generated by analyzing the logs of search queries that have been received from users (e.g., off-line, rather than in immediate response to a user's search query). Each logged search query can be considered in the same manner as described in reference to FIG. 1. The dictionary of common search queries can be used as a cache. When a search query is received from a user, the search query can be used to lookup corrections in the dictionary. If corrections exist in the dictionary, the search query can be corrected based on the dictionary entry. When a dictionary entry contains more than one potential correction (e.g., because the same search query could have been entered by more than one source keyboard layout), the source or target keyboard layout can be determined to resolve which dictionary correction to apply. If no corrections exist, the search query can be subject to the process 100 described in reference to FIG. 1.

[0055] Examples of the invention and all of the functional operations described in this specification can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Examples of the invention can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a computer-readable medium for execution by, or to control the operation of, data processing apparatus. The computer-readable medium can be a machine-readable storage device, a machine-readable storage substrate, a

memory device, a composition of matter effecting a machine-readable propagated signal, or a combination of one or more of them. The term "data processing apparatus" encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them. A propagated signal is an artificially generated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus.

[0056] A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub-programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[0057] The processes and logic flows described in this specification can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit).

[0058] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read-only memory or a random access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto-optical disks, or optical disks. However, a computer need not have such devices. Moreover, a computer can be embedded in another device, e.g., a mobile tele-

phone, a personal digital assistant (PDA), a mobile audio player, a Global Positioning System (GPS) receiver, to name just a few. Computer-readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[0059] To provide for interaction with a user, examples of the invention can be implemented on a computer having a display device, e.g., a CRT (cathode ray tube) or LCD (liquid crystal display) monitor, for displaying information to the user and a keyboard and a pointing device, e.g., a mouse or a trackball, by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input.

[0060] Examples of the invention can be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the invention, or any combination of one or more such back-end, middleware, or front-end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network ("LAN") and a wide area network ("WAN"), e.g., the Internet.

[0061] The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

[0062] While this specification contains many specifics, these should not be construed as limitations on the scope of the invention or of what may be claimed, but rather as descriptions of features specific to particular examples of the invention. Certain features that are described in this specification in the context of separate examples can also be implemented in combination in a single example. Conversely, various features that are described in the context of a single example can also be implemented in multiple examples separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more fea-

tures from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[0063] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various system components in the examples described above should not be understood as requiring such separation in all examples, and it should be understood that the described program components and systems can generally be integrated together in a single software product or packaged into multiple software products.

[0064] Thus, particular examples of the invention have been described. Other examples are within the scope of the following claims. For example, the actions recited in the claims can be performed in a different order and still achieve desirable results. Although the foregoing specification refers to input devices such as a keyboard which can have one of several keyboard layouts, the foregoing specification can equally apply to any device that can generate one of several sets of characters. Moreover, the Latin and Hebrew scripts used throughout the above specification are used only as examples; any other combination of keyboard layouts can be similarly used.

Claims

1. A computer-implemented method comprising:

receiving a first search query from a user (110), the first search query having one or more characters;
determining a first search result based on the first search query (120);
determining that the first search query has an incorrect input mapping (125), the determining being based on the number of items in the first search result;
identifying, for the first search query, a first keyboard layout used to provide the first search query (130);
identifying second keyboard layouts;
generating multiple corrected search queries from the first search query by mapping characters from the first keyboard layout to characters in the second keyboard layouts (160);
determining corrected search results based on the corrected search queries (170);
selecting one of the multiple corrected search queries;
comparing the number of items in the first search

result to the number of items in the search result based on the selected corrected search query; and
presenting to the user the search result with the greatest number of items (180).

2. The method of claim 1, wherein selecting one of the multiple corrected search queries further comprises presenting multiple potential search queries to the user for selection.

3. The method of claim 1, wherein receiving the first search query (110) comprises receiving a combination of two or more characters; and the method further comprises:

identifying a user language preference of the user using one or more of user languages settings, a user identified language, a user interface language, a geographic location, and user search history;
wherein identifying the second keyboard layout comprises identifying the second keyboard layout based on the user language preference of the user.

4. The method of claim 3, wherein the user language setting is associated with at least one of: a part of an operating system's language settings, information about a supplemental language support provided by an operating system, and information of a web browser.

5. The method of claim 4, wherein the information of the web browser is related to a language associated with the web browser.

6. The method of claim 3, wherein the user language preference is identified using the user language setting, and the method further comprises using a page request of a web browser to provide the user language setting.

7. The method of claim 3, wherein the identifying of the first keyboard layout (130) is based on a language setting or language information of a system used by the user.

8. The method of claim 3, wherein the identifying of the first keyboard layout (130) is based on primary and secondary languages identified by the user, or based on a user preferred keyboard layout.

9. The method of claim 3, wherein the user language preference is identified using the user interface language, and the identifying of the first keyboard layout is based on an inference from the user interface language.

10. The method of claim 3, wherein the identifying of the first keyboard layout (130) is based on search query history of the user, or based on information about a device connection of the user from which the user's geographic location and thus the user's most likely language are inferred. 5
11. The method of claim 1, wherein a portion of items identified by search results in a search query history, being less than an entire collection of items identified by search results in the search query history, is used to identify the first keyboard layout (130). 10
12. The method of claim 11, wherein the portion of items identified by search results in the search query history includes items that have been accessed by the user. 15
13. A computer program product, encoded on a computer readable medium, operable to cause data processing apparatus to perform operations comprising: 20

receiving a first search query from a user (110), the first search query having one or more characters; 25

determining a first search result based on the first search query (120);

determining that the first search query has an incorrect input mapping (125), the determining being based on the number of items in the first search result; 30

identifying, for the first search query, a first keyboard layout used to provide the first search query (130); 35

identifying second keyboard layouts;

generating multiple corrected search queries from the first search query by mapping characters from the first keyboard layout to characters in the second keyboard layouts (160); 40

determining corrected search results based on the corrected search queries (170);

selecting one of the multiple corrected search queries; 45

comparing the number of items in the first search result to the number of items in the search result based on the selected corrected search query; and

presenting to the user the search result with the greatest number of items (180). 50

14. A system, comprising:

means for receiving a first search query from a user (110), the first search query having one or more characters; 55

means for determining a first search result based on the first search query (120);

means for determining that the first search query has an incorrect input mapping (125), the determining being based on the number of items in the first search result;

means for identifying, for the first search query, a first keyboard layout used to provide the first search query (130);

means for identifying second keyboard layouts;

means for generating multiple corrected search queries from the first search query by mapping characters from the first keyboard layout to characters in the second keyboard layouts (160);

means for determining corrected search results based on the corrected search queries (170);

means for selecting one of the multiple corrected search queries;

comparing the number of items in the first search result to the number of items in the search result based on the selected corrected search query; and

means for presenting to the user the search result with the greatest number of items (180).

15. The system of claim 14, wherein means for selecting one of the multiple corrected search queries further comprise means for presenting multiple potential search queries to the user for selection.

30 Patentansprüche

1. Computerimplementiertes Verfahren, umfassend:

Empfangen einer ersten Suchanfrage von einem Benutzer (110), wobei die erste Suchanfrage ein oder mehrere Zeichen aufweist;

Bestimmen eines ersten Suchergebnisses auf der Grundlage der ersten Suchanfrage (120);

Bestimmen, dass die erste Suchanfrage eine inkorrekte Eingabebezuordnung aufweist (125), wobei das Bestimmen auf der Anzahl der Einträge im ersten Suchergebnis beruht;

Identifizieren eines ersten Tastaturlayouts für die erste Suchanfrage, das verwendet wurde, um die erste Suchanfrage bereitzustellen (160);

Identifizieren von zweiten Tastaturlayouts;

Erzeugen mehrerer korrigierter Suchanfragen aus der ersten Suchanfrage durch Zuordnen von Zeichen aus dem ersten Tastaturlayout zu Zeichen in den zweiten Tastaturlayouts (160);

Bestimmen von korrigierten Suchergebnissen auf der Grundlage der korrigierten Suchanfragen (170);

Auswählen von einer der mehreren korrigierten Suchanfragen;

Vergleichen der Anzahl der Einträge im ersten Suchergebnis mit der Anzahl der Einträge in dem Suchergebnis, das auf der ausgewählten

- korrigierten Suchanfrage beruht; und Präsentieren des Suchergebnisses mit der größten Anzahl von Einträgen für den Benutzer (180).
2. Verfahren nach Anspruch 1, worin das Auswählen von einer der mehreren korrigierten Suchanfragen ferner umfasst: Präsentieren mehrerer potenzieller Suchanfragen für den Benutzer zur Auswahl.
3. Verfahren nach Anspruch 1, worin das Empfangen der ersten Suchanfrage (110) das Empfangen einer Kombination von zwei oder mehr Zeichen umfasst; und das Verfahren ferner umfasst:
- Identifizieren einer Benutzersprachenpräferenz des Benutzers unter Verwendung von einem oder mehr von Folgendem: Benutzerspracheneinstellungen, eine durch den Benutzer gekennzeichnete Sprache, eine Sprache der Benutzeroberfläche, ein geografischer Standort und ein Benutzersuchverlauf; worin das Identifizieren des zweiten Tastaturlayouts umfasst: Identifizieren des zweiten Tastaturlayouts auf der Grundlage der Benutzersprachenpräferenz des Benutzers.
4. Verfahren nach Anspruch 3, worin die Benutzerspracheneinstellung mit mindestens einem von Folgendem assoziiert ist: ein Teil der Spracheinstellungen eines Betriebssystems, Information über eine durch ein Betriebssystem bereitgestellte ergänzende Sprachunterstützung und Information eines Webrowsers.
5. Verfahren nach Anspruch 4, worin die Information des Webbrowsers sich auf eine mit dem Webbrowser assoziierte Sprache bezieht.
6. Verfahren nach Anspruch 3, worin die Benutzersprachenpräferenz unter Verwendung der Benutzerspracheneinstellung identifiziert wird und das Verfahren ferner umfasst: Verwenden einer Seitenanforderung eines Webbrowsers, um die Benutzerspracheneinstellung bereitzustellen.
7. Verfahren nach Anspruch 3, worin das Identifizieren des ersten Tastaturlayouts (130) auf einer Spracheinstellung oder Sprachinformation eines durch den Benutzer verwendeten Systems beruht.
8. Verfahren nach Anspruch 3, worin das Identifizieren des ersten Tastaturlayouts (130) auf durch den Benutzer gekennzeichneten Primär- und Sekundärsprachen beruht oder auf einem durch den Benutzer bevorzugten Tastaturlayout beruht.
9. Verfahren nach Anspruch 3, worin die Benutzersprachenpräferenz unter Verwendung der Sprache der Benutzeroberfläche identifiziert wird und das Identifizieren des ersten Tastaturlayouts auf einer Herleitung aus der Sprache der Benutzeroberfläche beruht.
10. Verfahren nach Anspruch 3, worin das Identifizieren des ersten Tastaturlayouts (130) auf dem Suchanfrageverlauf des Benutzers beruht oder auf Information über eine Geräteverbindung des Benutzers, aus welcher der geografische Standort des Benutzers und somit die wahrscheinlichste Sprache des Benutzers hergeleitet wird, beruht.
11. Verfahren nach Anspruch 1, worin ein Anteil der durch Suchergebnisse in einem Suchanfrageverlauf identifizierten Einträge, welcher kleiner als eine ganze Sammlung von durch Suchergebnisse im Suchanfrageverlauf identifizierten Einträgen ist, verwendet wird, um das erste Tastaturlayout (130) zu identifizieren.
12. Verfahren nach Anspruch 11, worin der Anteil der durch Suchergebnisse identifizierten Einträge im Suchanfrageverlauf Einträge einschließt, auf die durch den Benutzer zugegriffen worden ist.
13. Computerprogrammprodukt, codiert auf einem computerlesbaren Medium und dafür betreibbar, eine Datenverarbeitungsvorrichtung zu veranlassen, Operationen durchzuführen, die Folgendes umfassen:
- Empfangen einer ersten Suchanfrage von einem Benutzer (110), wobei die erste Suchanfrage ein oder mehrere Zeichen aufweist; Bestimmen eines ersten Suchergebnisses auf der Grundlage der ersten Suchanfrage (120); Bestimmen, dass die erste Suchanfrage eine inkorrekte Eingabezuordnung aufweist (125), wobei das Bestimmen auf der Anzahl der Einträge im ersten Suchergebnis beruht; Identifizieren eines ersten Tastaturlayouts für die erste Suchanfrage, das verwendet wurde, um die erste Suchanfrage bereitzustellen (160); Identifizieren von zweiten Tastaturlayouts; Erzeugen mehrerer korrigierter Suchanfragen aus der ersten Suchanfrage durch Zuordnen von Zeichen aus dem ersten Tastaturlayout zu Zeichen in den zweiten Tastaturlayouts (160); Bestimmen von korrigierten Suchergebnissen auf der Grundlage der korrigierten Suchanfragen (170); Auswählen von einer der mehreren korrigierten Suchanfragen; Vergleichen der Anzahl der Einträge im ersten Suchergebnis mit der Anzahl der Einträge in dem Suchergebnis, das auf der ausgewählten

korrigierten Suchanfrage beruht; und
Präsentieren des Suchergebnisses mit der
größten Anzahl von Einträgen für den Benutzer
(180).

14. System, umfassend:

Mittel zum Empfangen einer ersten Suchanfrage
von einem Benutzer (110), wobei die erste
Suchanfrage ein oder mehrere Zeichen auf-
weist;

Mittel zum Bestimmen eines ersten Suchergeb-
nisses auf der Grundlage der ersten Suchanfrage
(120);

Mittel zum Bestimmen, dass die erste Suchanfrage
eine inkorrekte Eingabebezuordnung auf-
weist (125), wobei das Bestimmen auf der Anzahl
der Einträge im ersten Suchergebnis beruht;

Mittel zum Identifizieren eines ersten Tastatur-
layouts für die erste Suchanfrage, das verwen-
det wurde, um die erste Suchanfrage bereitzu-
stellen (130);

Mittel zum Identifizieren von zweiten Tastatur-
layouts;

Mittel zum Erzeugen mehrerer korrigierter
Suchanfragen aus der ersten Suchanfrage
durch Zuordnen von Zeichen aus dem ersten
Tastaturlayout zu Zeichen in den zweiten Tas-
taturlayouts (160);

Mittel zum Bestimmen von korrigierten Sucher-
gebnissen auf der Grundlage der korrigierten
Suchanfragen (170);

Mittel zum Auswählen von einer der mehreren
korrigierten Suchanfragen;

Vergleichen der Anzahl der Einträge im ersten
Suchergebnis mit der Anzahl der Einträge in
dem Suchergebnis, das auf der ausgewählten
korrigierten Suchanfrage beruht; und

Mittel zum Präsentieren des Suchergebnisses
mit der größten Anzahl von Einträgen für den
Benutzer (180).

15. System nach Anspruch 14, worin die Mittel zum Aus- wählen von einer der mehreren korrigierten Such- anfragen ferner Mittel zum Präsentieren mehrerer potenzieller Suchanfragen für den Benutzer zur Aus- wahl umfassen.

Revendications

1. Procédé mis en oeuvre par ordinateur comprenant :

la réception d'une première requête de recher-
che en provenance d'un utilisateur (110), la pre-
mière requête de recherche comportant un ou
plusieurs caractère(s) ;

la détermination d'un premier résultat de recher-
che sur la base de la première requête de re-
cherche (120) ;

la détermination du fait que la première requête
de recherche présente une cartographie d'en-
trée incorrecte (125), la détermination étant ba-
sée sur le nombre d'éléments dans le premier
résultat de recherche ;

l'identification, pour la première requête de re-
cherche, d'une première configuration de clavier
qui est utilisée pour fournir la première requête
de recherche (130) ;

l'identification de secondes configurations de
clavier ;

la génération de multiples requêtes de recher-
che corrigées à partir de la première requête de
recherche en cartographiant des caractères en
provenance de la première configuration de cla-
vier selon des caractères conformément aux se-
condes configurations de clavier (160) ;

la détermination de résultats de recherche cor-
rigés sur la base des requêtes de recherche cor-
rigées (170) ;

la sélection de l'une des multiples requêtes de
recherche corrigées ;

la comparaison du nombre d'éléments dans le
premier résultat de recherche avec le nombre
d'éléments dans le résultat de recherche qui est
basé sur la requête de recherche corrigée
sélectionnée ; et

la présentation à l'utilisateur du résultat de re-
cherche qui présente le nombre le plus impor-
tant d'éléments (180).

2. Procédé selon la revendication 1, dans lequel la sé-
lection de l'une des multiples requêtes de recherche
corrigées comprend en outre la présentation de mul-
tiples requêtes de recherche potentielles à l'utilisa-
teur dans le but d'une sélection.

3. Procédé selon la revendication 1, dans lequel la ré-
ception de la première requête de recherche (110)
comprend la réception d'une combinaison de deux
caractères ou plus ; et le procédé comprend en
outre :

l'identification d'une préférence de langue d'uti-
lisateur de l'utilisateur en utilisant l'un ou plu-
sieurs parmi des définitions de langues d'utili-
sateur, une langue identifiée d'utilisateur, une
langue d'interface d'utilisateur, une localisation
géographique et un historique de requêtes de
recherche d'utilisateur ;

dans lequel l'identification de la seconde confi-
guration de clavier comprend l'identification de
la seconde configuration de clavier sur la base
de la préférence de langue d'utilisateur de l'uti-
lisateur.

4. Procédé selon la revendication 3, dans lequel la définition de langue d'utilisateur est associée à au moins l'un parmi : une partie de définitions de langue de système d'exploitation, une information qui concerne un support de langue supplémentaire qui est fourni par un système d'exploitation et une information d'un navigateur Web.
5. Procédé selon la revendication 4, dans lequel l'information du navigateur Web est rapportée à une langue qui est associée au navigateur Web.
6. Procédé selon la revendication 3, dans lequel la préférence de langue d'utilisateur est identifiée en utilisant la définition de langue d'utilisateur, et le procédé comprend en outre l'utilisation d'une requête demandant une page d'un navigateur Web pour fournir la définition de langue d'utilisateur.
7. Procédé selon la revendication 3, dans lequel l'identification de la première configuration de clavier (130) est basée sur une définition de langue ou une information de langue d'un système qui est utilisé par l'utilisateur.
8. Procédé selon la revendication 3, dans lequel l'identification de la première configuration de clavier (130) est basée sur des langues primaire et secondaire qui sont identifiées par l'utilisateur, ou est basée sur une configuration de clavier préférée d'utilisateur.
9. Procédé selon la revendication 3, dans lequel la préférence de langue d'utilisateur est identifiée en utilisant la langue d'interface d'utilisateur et l'identification de la première configuration de clavier est basée sur une déduction résultant de la langue d'interface d'utilisateur.
10. Procédé selon la revendication 3, dans lequel l'identification de la première configuration de clavier (130) est basée sur un historique de requêtes de recherche de l'utilisateur ou est basée sur une information qui concerne une connexion de dispositif de l'utilisateur à partir de laquelle la localisation géographique d'utilisateur et par conséquent, la langue la plus probable d'utilisateur sont déduites.
11. Procédé selon la revendication 1, dans lequel une section d'éléments qui sont identifiés au moyen de résultats de recherche dans un historique de requêtes de recherche, d'une dimension plus petite qu'une collection entière d'éléments qui sont identifiés au moyen de résultats de recherche dans l'historique de requêtes de recherche, est utilisée pour identifier la première configuration de clavier (130).
12. Procédé selon la revendication 11, dans lequel la section d'éléments qui sont identifiés au moyen de résultats de recherche dans l'historique de requêtes de recherche inclut des éléments qui ont été accédés par l'utilisateur.
13. Progiciel, codé sur un support lisible par ordinateur, opérationnel pour amener un appareil de traitement de données à réaliser les opérations comprenant :
- la réception d'une première requête de recherche en provenance d'un utilisateur (110), la première requête de recherche comportant un ou plusieurs caractère(s) ;
la détermination d'un premier résultat de recherche sur la base de la première requête de recherche (120) ;
la détermination du fait que la première requête de recherche présente une cartographie d'entrée incorrecte (125), la détermination étant basée sur le nombre d'éléments dans le premier résultat de recherche ;
l'identification, pour la première requête de recherche, d'une première configuration de clavier qui est utilisée pour fournir la première requête de recherche (130) ;
l'identification de secondes configurations de clavier ;
la génération de multiples requêtes de recherche corrigées à partir de la première requête de recherche en cartographiant des caractères en provenance de la première configuration de clavier selon des caractères conformément aux secondes configurations de clavier (160) ;
la détermination de résultats de recherche corrigés sur la base des requêtes de recherche corrigées (170) ;
la sélection de l'une des multiples requêtes de recherche corrigées ;
la comparaison du nombre d'éléments dans le premier résultat de recherche avec le nombre d'éléments dans le résultat de recherche qui est basé sur la requête de recherche corrigée sélectionnée ; et
la présentation à l'utilisateur du résultat de recherche qui présente le nombre le plus important d'éléments (180).
14. Système, comprenant :
- des moyens pour recevoir une première requête de recherche en provenance d'un utilisateur (110), la première requête de recherche comportant un ou plusieurs caractère(s) ;
des moyens pour déterminer un premier résultat de recherche sur la base de la première requête de recherche (120) ;
des moyens pour déterminer que la première requête de recherche présente une cartographie d'entrée incorrecte (125), la détermination

- étant basée sur le nombre d'éléments dans le premier résultat de recherche ;
des moyens pour identifier, pour la première requête de recherche, une première configuration de clavier qui est utilisée pour fournir la première requête de recherche (130) ; 5
des moyens pour identifier des secondes configurations de clavier ;
des moyens pour générer de multiples requêtes de recherche corrigées à partir de la première requête de recherche en cartographiant des caractères en provenance de la première configuration de clavier selon des caractères conformément aux secondes configurations de clavier (160) ; 10
des moyens pour déterminer des résultats de recherche corrigés sur la base des requêtes de recherche corrigées (170) ;
des moyens pour sélectionner l'une des multiples requêtes de recherche corrigées ; 20
comparer le nombre d'éléments dans le premier résultat de recherche avec le nombre d'éléments dans le résultat de recherche qui est basé sur la requête de recherche corrigée sélectionnée ; et 25
des moyens pour présenter à l'utilisateur le résultat de recherche qui présente le nombre le plus important d'éléments (180).
15. Système selon la revendication 14, dans lequel les 30
moyens pour sélectionner l'une des multiples requêtes de recherche corrigées comprennent en outre des moyens pour présenter de multiples requêtes de recherche potentielles à l'utilisateur dans le but d'une sélection. 35

40

45

50

55

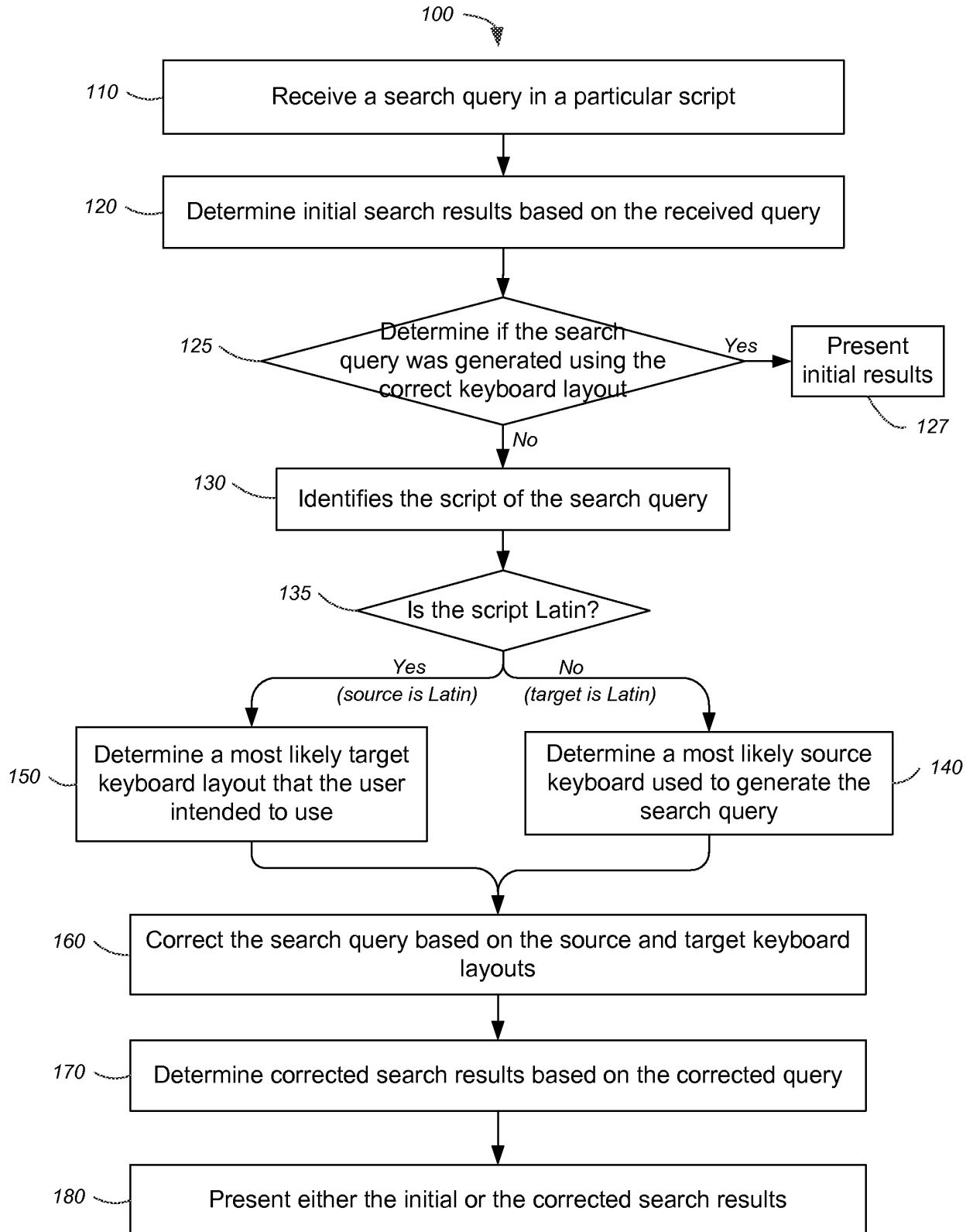


FIG. 1

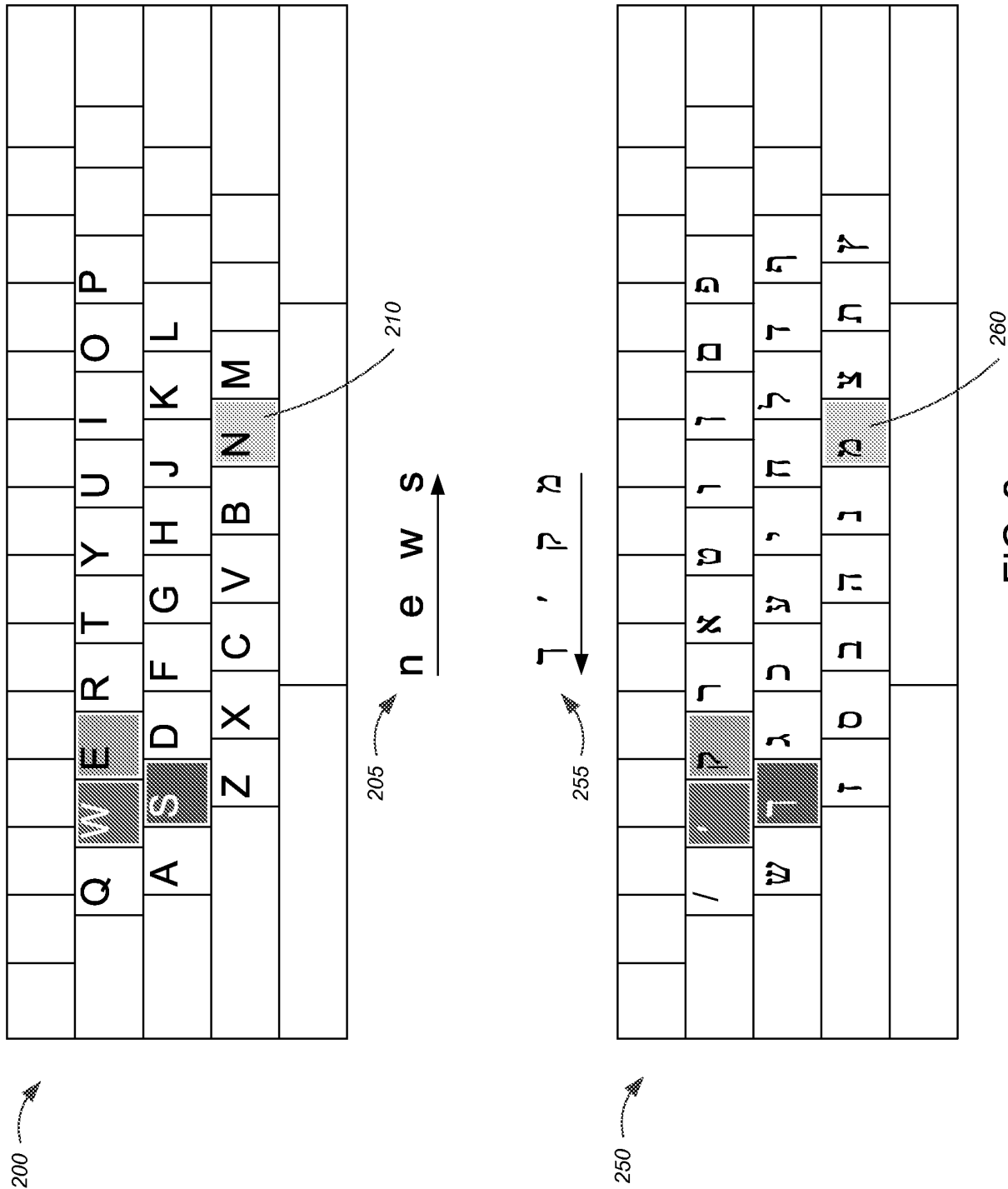


FIG. 2

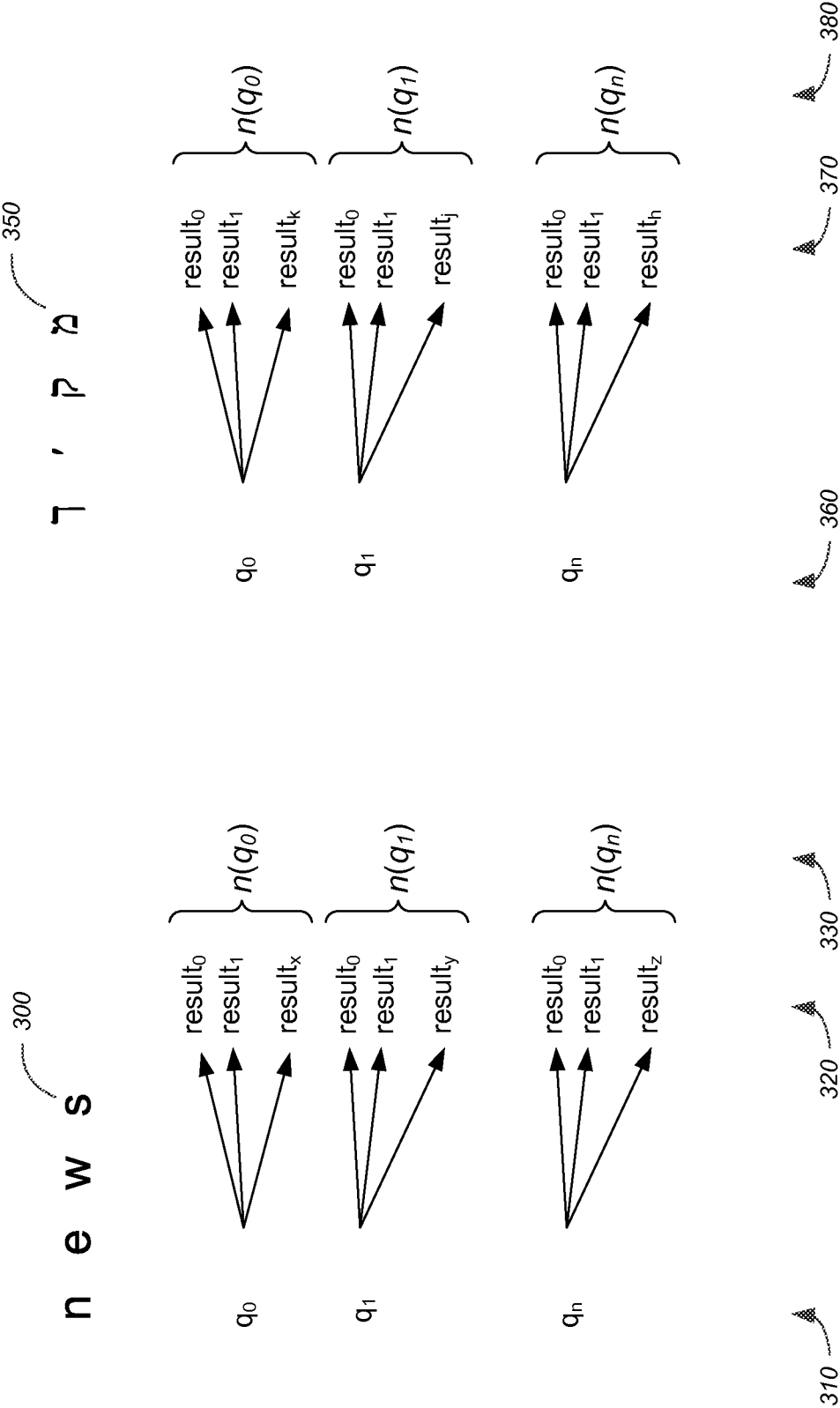


FIG. 3

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- CN 200710101817 [0001]
- US 87542506 P [0001]
- US 88483907 P [0001]
- EP 0967555 A2 [0005]
- US 2005071332 A1, ORTEGA RUBEN ERNESTO [0005]